I'm not robot
reCAPTCHA
Continue

La comunicacion y sus ellas para niños

numpy.identity(n, dtype - None) : Devuelve la matriz de identidad, es decir. Parámetros : n : [int] Dimension n x n output field dtype : [optional, float(by Default)] Tipo de datos del campo devuelto. Devuelve : una matriz de dimensión de identidad n x n, con su diagonal principal establecida en uno y todos los demás elementos 0. Ejemplo: import numpy como geek b á geek.identity(2, dtype á float) print(Matrix b : , b) y ?geek.identity(4) print(Matrix a : , a) Salida : Matrix b : [[1. 0.] [0. 1.]] Matriz y : [[1. 0. 0. 0.] [0. 1. 0. 0.] [0. 0. 1. 0.] [0. 0. 0. 1.]] Nota: Estos códigos no se ejecutarán en el ID en línea. Por favor, ejecútelos en sus sistemas para explorar el trabajo. Este artículo es contribuido por Mohit Gupta_OMG 😀. Si te gusta GeeksforGeeks y te gustaría contribuir, también puedes escribir un artículo usando contribute.geeksforgeeks.org o enviar tu artículo a contribute@geeksforgeeks.org. Echa un vistazo a tu artículo para que aparezca en la página principal de geeksforGeeks y ayuda a otros Geeks.Por favor, publica comentarios si encuentras algo mal o quieres compartir más información sobre el tema discutido anteriormente. ¡Atención geek! Fortalezca sus conceptos básicos con el curso Python Programming Foundation y aprenda los conceptos básicos. En primer lugar, los preparativos de las entrevistas mejoran los conceptos de la estructura de datos con el tutorial de Python DS. Publicaciones destacadas: Mejorado: anshitaagarwal En este tutorial aprenderemos a imprimir una matriz de identidad en python. Una matriz se denomina matriz de identidad si todos sus elementos diagonales están desde la esquina superior izquierda hasta la esquina inferior derecha de 1, y todos los demás elementos son 0. Por ejemplo, las matrices siguientes son matrices de identidad: las tres matrices constan de ceros excepto diagonales. Los elementos para diagonales desde la esquina superior izquierda hasta la esquina inferior derecha son uno. Los otros dos términos que necesitamos saber sobre la matriz son 'fila' y 'columna'. Si la matriz contiene las filas m y n columnas, se denomina matriz (m x n). Tenga en cuenta que la matriz de identidad también se conoce como la matriz de unidades. El número de filas es igual al número de columnas para la matriz de identidad. Por lo tanto, en lugar de fila y columna, las matrices de identidad se definen por su tamaño. Si el tamaño es 'n', tendrá un número 'n' de filas y un número 'n' de columnas. Estos tipos de matrices con el mismo tamaño de fila y columna también se denominan matrices cuadradas. En este tutorial, aprenderemos a imprimir una matriz de identidad mediante python. En nuestro programa, pedimos al usuario que escriba el tamaño de la matriz y, a continuación, imprima la matriz de identidad. Python : tamaño á int(input(Enter matrix size :)) for even in the range(0,size): for j in the range(0,size): if(i-j): print(1,end -) else print: (0,end á) print() Código fuente está disponible aquí. Explicación : La idea principal de imprimir la matriz de identidad que su valor será igual a 1 si el número de fila actual es igual al número de columna actual. Para la primera fila, la matriz es 1, para la segunda fila la matriz es 1, y así sucesivamente. Por lo tanto, para imprimir la matriz, usamos dos bucles, y si ambos contadores son los mismos para los bucles, imprima 1. De lo contrario, imprima 0. Los números anotados en el programa anterior indican el número de paso a continuación : Pida al usuario que introduzca el tamaño de la matriz y guárdela en un tamaño variable. Este será el tamaño de fila y columna para la matriz de identidad. Ejecute un bucle de 0 al tamaño de la matriz - 1. Lo usamos para un bucle para imprimir filas de tuercas. Si el valor actual en este bucle es 0, significa que estamos trabajando en la primera fila de la matriz, si es 1, estamos trabajando en la segunda fila de la matriz, etc. Inicie un bucle interno más y ejecútelo desde 0 hasta el tamaño de la tuerca - 1. Este bucle interno se utiliza para trabajar en las columnas de la tuerca. Si el valor actual en este bucle es 0, trabajamos en la primera columna de la matriz, si es 1, trabajamos en la segunda columna de la matriz, y así sucesivamente. Por lo tanto, para cualquier valor del bucle externo, es decir, el bucle en el paso 2, este bucle interno se ejecutará completamente. primero imprimimos todos los valores de columna para la primera fila, luego para la segunda fila, etc. Compruebe if_ i_ es el to_ j_ o no. Si es así, imprima 1, de lo contrario imprima 0. Este paso se utiliza para imprimir 1 en una matriz diagonal. Imprimimos 1, solo si el índice actual de la fila de bucle externo es igual a la columna de índice del bucle interno. Imprima una nueva línea. O ir a la siguiente línea. Salida de muestra : Tutoriales similares : Requisitos: 先決条件: En el álgebra lineal, la matriz de identidad es n n × n matriz cuadrada con los de la diagonal principal y ceros en otro lugar. También está marcado. También se conoce como una matriz de unidades porque su valor determinante es 1 independientemente del tamaño. Esta es una característica clave de la matriz de identidad y juega un papel importante en el álgebra lineal. El 线性代数单位矩阵是为单位矩阵是n阶方阵，其他方阵，用表示. 也称为单位矩阵,因为其行列式值为1,与大小无关。 这是恒等矩阵的关键特征.在线性代数中起着重要的作用。 La matriz de identidad tiene una propiedad que la multiplicación a la matriz de identidad proporciona una matriz de identidad (ik -I). 单位矩阵以下性质:将k个单位矩阵乘得到个单位矩阵(I k á 1). 代码性份矩阵 #Linear Secuencia de aprendizaje de álgebras (Identity Matrix Property (I-k - I) import numpy as np - identity Matrix I ? np.eye(4) print(---I(4x4)---, I) k á 14 Ik - I para y en el rango(14): Ik á I*Ik print('--- I-k ----', Ik) Salida: --- 輸(4x4)--- [[1]. 0. 0. 0.] [0. 1. 0. 0.] [0. 0. 1. 0.] [0. 0. 0. 1.]] --- I-k ---- [[1. 0. 0. 0.] [0. 1. 0. 0.] [0. 0. 1. 0.] [0. 0. 0. 1.]] Traducido de: Likes Collections Report Watch One Click Three Even as Mark Follow Blogger, Manténgase al día con la última entrada de blog ©2020 CSDN Skin Theme: Programming Studio Designer: CSDN Official Blog Back Home Python Build Symmetry Matrix Assumptions: Defining Matrix Definition Matrix Identity Matrix Transpose Matrix In Linear Algebra, si las matrices de su transposición son las mismas, entonces la matriz es simétrica (MT con M) Para los elementos de matriz: M (i, j) con M (j, i) por elementos de matriz: M (i,j) con M (j,i) Seguido de código python para desarrollar el control de matriz simétrica. Aquí hay un código Python que muestra cómo examinar la matriz de simetría. Método: Sintaxis: M snmpy.array () transpose_M con M.T si transpose_M con M: Transpona con True: código Python para matriz simétrica SM (código Python para matrices simétricas) con aprendizaje de aprendizaje de álgebra lineal con Transposición utilizando varios métodos para importar asnp asnp asnpy numpy. (3.45, 8, 78) impresión (---Matrix M---, M) con impresión Matrix M (Transponer como M.T----, M.T., si M.T.all () - M.all(): impresión (--------> Transponer es equal a M) M - np.array ('2,3,4', Impresión '3,45,8') (---Matrix B---, M) con transposición de la impresión matriz M ('Transposición como B.T----', M.T.) si M.T.all () con M.all() y np.shape (M) snp.shape (M.T): print ('---------> Transpose is equal to B) else ('---------> Trans Notose!! Salida: --- Matrix M--- (2 3 4) (3 45 8) . . . . . ---------> ---

. . . . . Traducido de: Python genera Matrix Symmetry Likes Collections Siga el triple de un clic como Mark para seguir al blogger, manténgase al día con el último blog de la ©2020 CSDN Skin Theme: Programming Studio Designer: CSDN Official Blog Return Home Task Build Matrix Size en t

ejecución. Una matriz de identidad es una matriz cuadrada de tamaño n × n, donde los elementos diagonales son todos 1s (unos) y todos los demás elementos son todos 0s (ceros). I n ..... I_ ⋮ ⋮ ⋮ ⋱ .. ! .....

. . . . . vdots s/ddots s/vdots

. . . . . s/ s/Ulam_spiral_ (for_primes) 11(editar) F identity_matrix (tamaño) <

In.....

. . . . .

řádek) identity_matrix (3) tisk (řádek) (1, 0, 0) (0, 1, 0) . . . . .

USING INDENMAT,R13 base register B 72(R15) skip savearea DC 17F'0' savearea STM R14,R12,12(R13) save previous context ST R13,04(R15) enlace hacia atrás ST R15,8 (R13) enlace directo LR R13, R15 set direccionability L R1,N n MH R1,N+2 n*n SLA R1,2 *4 ST R1, LL cantidad de almacenamiento requerido por GETMAIN RU, EL LV (R1) asigna el almacenamiento para las matrices DE DYNA, R11, para hacer que el almacenamiento direccionable LR R11, R1 establezca la direccionamiento LA R6,1 i a 1 DO WHILE (C, R6,LE,N) a i-1 a n LA R7,1 j-1 DO WHILE (C,R7,LE,N) a j-1 a n IF CR,R6,EQ,R7 Then si i=j entonces LA R2-j ,1 k-1 ELSE , de lo contrario LA R2.0 k-0 ENDIF , endif LR R1,R6 i BCTR R1.0 -1 MH R1,N+2 *n AR R1,R7 (i-1)*n +j BCTR R1.0 -1 SLA R1.2 *4 ST R2,A(R1) (i,j)-k LA R7.1(R7) j++ ENDDO , enddo j LA R6.1(R6) i++ ENDDO , enddo i LA R6.1 i-1 DO WHILE (C,R6,LE,N) a i-1 a n LA R10.PG pgi-0 LA R7.1 j-1 DO WHILE-WHILE-WHILE(WHILE- WHILE- WHILE- C,R7,LE,N) hasta j-1 a n LR R1,R6 i BCTR R1.0 -1 MH R1,N+2 *n AR R1,R7 (i-1)*n+j BCTR R1 .0 -1 SLA R1.2 *4 R2,A(R1) e (i,j) XDECO R2, Edición XDEC MVC 0(1,R10),XDEC+11 salida LA R10.1(R10) pgi++R10) a 1 LA R7.1 (R7) j ++ ENDDO , enddo j XPRNT PG, L'PG impresión LA R6.1 (R6) i ++ ENDDO, enddo i LA R1, dirección libre LA R2, importe de almacenamiento libre LL FREEMAIN A á (R1), LV (R2) almacenamiento asignado libre DROP R11 drop register L R13,4(0,R13) restaurar el indicador de savearea anterior LM R14, R12,12 (R13) para restaurar el contexto anterior XR R15, R15 rc a 0 BR R14 exitNN EQU 10 parámetro n (90->32K)N DC A(NN) nLL DS F n*n*4PG DC CL(NN)' ' bufferXDEC DS CL12 tempDYNA DSECTA DS F A(n n ,n) YREGS END INDENMAT 1000000000 0100000000000 001000000 000100000 000000 000000 000000000 000001000 000000000000 000000010 000000001 Ada[editar] Cuando se utilizan matrices de punto flotante en Ada 2005+, la función se define como Unit_Matrix en Ada.Numerics.Generic_Real_Arrays. Como paquete genérico, puede trabajar con tipos definidos por el usuario con decimales flotantes o con Short_Real_Arrays, Real_Arrays y Long_Real_Arrays integrados. Como se puede ver a continuación, también se pueden establecer los primeros índices de ambas dimensiones, porque los índices del campo Ada no comienzan arbitrariamente con un número específico. -- Como se prototipo en la Generic_Real_Arrays:-- función Unit_Matrix (Orden: Positivo; First_1, First_2 : Integer : 1) devuelve Real_Matrix;-- Para task:mat : Real_Matrix : à Unit_Matrix(5); Para versiones anteriores de Ada o sin punto flotante, sus tipos vuelven a lo básico: el tipo Matrix es una matriz (Rango positivo <>, Rango positivo <>) Entero;mat: Matrix(1.5.1..5) :- (otros ?> 0);-- y luego después de las partes declarativas: incluso en marTrange(3) loop mat(i,i) bucle final; ALGOL 68[editar] Nota: El código vectorial genérico y de matriz debe moverse a una página más general. Archivo: prelude/vector_base.a68-l/usr/bin/a68g --script -*- codificación: utf-8 -*- ? Definir algunas operaciones generales de inicialización e impresión de vectores - COMMENT MODE SCAL á skalár, siesta REAL; FORM-T ákalování fmt ;; FINAL COMENTARIO INT vec lwb :- 1, vec upb :- ;- VECNEW á [vec lwb:vec upb]SCAL; RE-IM VECT - REF VECNEW; FORMAT vec fmt :- $(n{vec upb-vec lwb}f)(scal fmt),)f(scal fmt),)f(scal fmt)$; PRIO INIT n,o 1; OP INIT (VEC self, SCAL scal)VEC: (PARA el col DE LWB self á UPB self[col]:- scal OD; self); • ZEROINIT: definuje aditivní identitu #OP ZEROINIT (VEC self)VEC: vlastní INIT SCAL(O); OP REPR (VEC self)STRING: (SOUBOR f; STRING s; spolupracovník(f,s); vec lwb :- LWB vlastní; vec upb :- UPB self; put(f, (vec fmt, self)); zav'it(f); s); SKIPFile: prelude/matrix_base.a68o -*- kódování: utf-8 -*- á Definovat n'ové generické matice inicializace a tiskové operace ? KOMENT. RE-IM VEC - [SCAL; FORM-T ákalování fmt ;; END COMMENT INT mat lwb :- 1, mat upb :- 0; MODE MATNEW á [mat lwb:mat upb, vec lwb: vec upb]SCAL; ALFOMBRA DE REIM - REF MATNEW; FORM-T mat fmt á $(n{vec upb-vec lwb}f(vec fmt),)x(f(vec fmt))$; PRIO DIAGINIT n,o 1; OP INIT (REF mat self, SCAL scal)MAT: (pro ádek z LWB self TO UPB self['ádek,] INIT scal OD;self); • ZEROINIT: definuje aditivní identitu #OP ZEROINIT (MAT self)MAT: self INIT SCAL(O); OP REPR (MATNEW self)STRING: (SOUBOR f; STRING s; spolupracovník(f,s); vec lwb :- 2 LWB vlastní; vec upb : 2 UPB vlastní; estera lwb :- LWB self; mat upb :- UPB self; put(f, (mat fmt, self)); zav'it(f); s); OP DIAGINIT (MAT já, VEC diag)MAT: (ZEROINIT já; PARA D DE LWB diag A UPB diag DO self[d,d]:- diag[d] OD;- nebo alternativn- pomocí TORRIX ... DIAG self:- diag;- self); • ONEINIT: definuje multiplikativní identitu #OP ONEINIT (MAT self)MAT: (ZEROINIT vlastní DIAGINIT (LOC[LWB self:UPB self]SCAL INIT SCAL(1)) - nebo alternativn-pomocí TORRIX ... (DIAG ven) VECINIT SCAL(1)-); SKIPFile: prelude/matrix_ident.a68o -*- kódování: utf-8 -*- - PRIO INTERT -9; • Stejn-jako já pro COMPLex - OP IDENT (INT lwb, upb)MATNEW: ONEINIT LOC [lwb:upb,lwb:upb]SCAL; OP IDENT (INT upb)MATNEW: á v-chozí lwb je 1 a 1 IDENT upb; SKIPFile: prelude/matrix_a68-l/usr/bin/a68g --script-*- kódování: utf-8 -*- - PR READ prelude/vector_base.a68 PR; PR READ prelude/matrix_base.a68 PR; PR READ prelude/matrix_ident.a68 PR; SKIPFile: test/matrix_ident.a68-l/usr/bin/a68g --script-*- kódování: utf-8 -*- - MODE SCAL - REAL; FORMAT scal fmt :- $g(-3,1)$; PR READ prelude/matrix.a68 PR; (REPR IDENT 4) ((1,0, 0,0,0, 0,0, 0,0, 0,0), (0,0, 1,0, 0,0, 0,0), (0,0, 0,0, 1,0, 0,0), (0,0, 0,0, 0,0, 1,0), (0,0, 0,0, 0,0, 0,0, 1,0), (0,0, 0,0, 0,0, 0,0, 0,0, 1,0)) ALGOL W[editar] zasíná % nastaveno m na matici identity velikosti s % procedury makeIdentity(skute-né pole m (*, *) ; celo-iselná hodnota s) ; pro i :o 1, dokud neza-ne s pro j : 1, dokud s do m(i, j) : à 0,0; m(i, i) : 1.0 extrema makeIdentity ; % prueba makeIdentity postup % za-it skute-né pole id5(1 :: 5, 1 :: 5); makeIdentity(id5, 5); r_format: A r_w: 6; r_d: 1; % nastavená v-stupní formát pro reals % pro i :' 1 a a à 5 za-inaji psát(); j : 1 a 5 a writeln(id5(i, j) termina con for_i; fin del texto. APL[editar] La creación de una matriz de identidad en la API, incluye el producto externo de la función de igualdad. Para la matriz cuadrada 3: ⍳¨⍳←⍳3 ID 51 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 Puede definir una función silenciosa mediante una de las siguientes líneas equivalentes: ID←{⍵ ⍵∘.=⍵} ⍝ ID ⍵ ⍝ ⍵ ⍵∘.=⍵ ⍝ ⍵ ⍵∘.=⍵ ⍝ ⍵ ⍵∘.=